THE APPLICATION STUDIES OF EMBEDDED SYSTEM DEVELOPMENT PLATFORMS

Guoping Wang

Department of Engineering,
Indiana University Purdue University Fort Wayne, Fort Wayne, Indiana 46805
Email: wang@ipfw.edu

1. ABSTRACT

Embedded computer systems are changing more often than other computing environments since the scope of their application domain is expanding. In the past, embedded system development was largely focused on 8-bit, standalone systems written directly in assembly language. These systems were characterized by slow CPUs (Central Processing Unit), kilobytes of memory, and limited peripheral devices on the chip. Nowadays they are embracing ever-widening application domains to include not only 8-bit standalone systems but real-time, networked, Operating-System based, wireless systems with megabytes of memory and 32-bit CPUs, with very rich peripheral devices. In the same time, the rapid advancement of low-cost embedded system kits such as Raspberry Pi[1], Arduino[2], etc. propose challenges and also opportunities to engineering educators for delivering contents and skills and engaging students in hands-on projects in embedded system teaching, which may take the advantages of a variety of platforms. On one hand, there are so many embedded platforms to choose, from ARM-based development kits, such as, MCB Series from Keil, Lab-in-Box from FreeScale, LaunchPad from TI, etc to low-cost platforms such as Arduino, Raspberry Pi. These platforms can help educators and students to accelerate their learning pace by working on real-world projects. On the other hand, this also proposes challenges as students are confused to pick which platform for their design projects. In this paper, three main ARM-based embedded system platforms are studied, and, the features and their application areas are compared. This study will be very helpful in the teaching and learning of embedded systems to engineering educators and students and give students some ideas to find suitable embedded system platforms in their design projects.

2. INTRODUCTION

An embedded system is a special-purpose computer system in which the computer is completely encapsulated by the device it controls. Unlike a general-purpose computer, such as a personal computer, an embedded system performs pre-defined tasks, usually with very specific requirements. Since the system is dedicated to a specific task, design engineers can optimize it, reducing the size and cost of the product. Embedded systems are often mass-produced, so the cost savings may be multiplied by millions of items.

Some examples of embedded systems include ATMs, cell phones, printers, thermostats, calculators, and videogame consoles. Handheld computers or PDAs are also considered embedded devices because of the nature of their hardware design, even though they are more expandable in software terms. This line of definition continues to blur as devices expand.

A microcontroller (sometimes abbreviated µC, uC or MCU) is a small computer on a single integrated circuit containing a processor core, memory, and programmable input/output peripherals. Program memory in the form of flash is also often included on chip, as well as a typically small amount of RAM (Read Access Memory). Microcontrollers are designed for embedded applications, in contrast to the microprocessors used in personal computers or other general purpose applications.

As the heart of an embedded system, the microcontroller has seen significant advancements in the last thirty years, from 8-bit to 32-bit, from very limited on-chip resources to very rich peripherals among which you can find almost everything needed for embedded applications nowadays.

MCS-8051, one example of early 8-bit MCU, was introduced by Intel Corporation in 1981[3]. This MCU has only 128 bytes of RAM, 4K bytes of on-chip ROM (Read Only Memory), two timers, one serial port, and four ports (each 8-bit wide) all on a single chip. Intel's original 8051 versions were popular in the 1980s and early 1990s and enhanced binary compatible derivatives remain popular today.

From 1996, the 8051 family MCU was continued with enhanced 8-bit MCS-151 and the 8/16/32-bit MCS-251 family MCUs. While Intel no longer manufactures the MCS-51, MCS-151 and MCS-251 family, enhanced binary compatible derivatives made by numerous vendors remain popular today. These enhanced MCUs also embed many peripherals on-chip, such as flash RAM, A/D and D/A converters, Real-time Clock, watch dog, even USB on the chip. Those MCS-51 compatible MCUs find popular applications in low-end embedded systems.

In the last ten years, along with the surge of mobile devices, 32-bit ARM processor has been very popular. In 2005, about 98% of all mobile phones sold used at least one ARM processor[4]. The low power consumption of ARM processors has made them very popular today. As of 2013, 37 billion ARM processors have been produced, up from 10 billion in 2008[5]. The ARM architecture is the most widely used architecture in mobile devices, and most popular 32-bit MCU in embedded systems[6]. Today, one of the popular ARM architecture Cortex-M3 chip includes numerous peripherals such as General Purpose I/O ports, timers, PWM, UART, SPI, I^2C, A/D, D/A, CAN, USB, Ethernet and on-chip flash memory programming, etc.

Recently, low-cost credit-size embedded system kits are becoming very popular among hobbies, as well as engineering students' design projects. For example, Arduino platforms are used in robots, thermostats, motion detectors, etc. Raspberry Pi platforms have found applications in personal multi-media projects, Wi-Fi enabled home surveillance system, etc. In the classroom of embedded system teaching, ARM-based development boards are still widely used. The emergence of these low-cost embedded system kits is helping educators and students to accelerate their learning pace by working on real-world projects. However, it also proposes challenges as students are confused to choose which platform they can use in their design projects. In this paper, three main ARM-based embedded system platforms are studied, and, the features and their applicable domains are compared. The comparison results can help students and educators to a better idea about the pros and cons of each platform and their suitable application areas.

3. EMBEDDED SYSTEM PLATFORMS APPLICATION STUDIES

In this section, first, we look at two very popular embedded system platforms, Arduino and Raspberry Pi, and then the ARM-based C language supported development boards are studied.

Arduino is a single-board MCU platform intended to make the application of interactive objects or environments more accessible. Typically, the hardware consists of an open-source hardware board using an 8-bit Atmel AVR MCU, or a 32-bit ARM MCU. Arduino board can also interact with their environment using boards with sensors, actuators, GUI interface, wireless modules, etc. It also comes with a simple IDE that runs on regular PC and allows users to write programs for Arduino using C or C++.

An Arduino board consists of an Atmel 8-bit AVR MCU or an 32-bit Atmel ARM MCU with complementary components. An important feature of the Arduino is its standard connectors, which lets users connect to various Arduino accessory boards (called shields)[7]. Some shields communicate with the Arduino board directly or through I^2C serial bus and many shields can be stacked and used in parallel. The Arduino shields can provide functions such as motor controls, wireless communication XBee, GPS, Ethernet, LCD display, etc. The software development of Arduino, typically include an Integrated Development Environments (IDE) and Arduino libraries for various shields and Arduino programs are written in C or C++. Here are some typical projects using Arduino, a temperature/pressure sensing/controller system with GUI display, a robotic football team, etc.

The Raspberry Pi is a credit card-sized single-board computer developed in the UK by the Raspberry Pi Foundation with the intention of promoting the teaching of basic computer science in schools. The Raspberry Pi has a Broadcom BCM2835 system on a chip (SoC), which includes an ARM1176JZF-S 700 MHz processor, VideoCore IV GPU, and was originally shipped with 256 megabytes of RAM, later upgraded (Model B & Model B+) to 512 MB. It does not include a built-in hard disk or solid-state drive, but it uses an SD card for booting and persistent storage, with the Model B+ using a MicroSD. The Foundation provides Debian and Arch Linux ARM distributions for download. The Raspberry PI can be easily connected to a computer keyboard, LCD monitor and Internet. Tools are available for Python as the main programming language, with support for BBC BASIC(via the RISC OS image or the Brandy Basic clone for Linux), C, Java, Perl and Ruby. However, the Input/Output pins of Raspberry Pi is very limited, thus, the Raspberry Pi differentiates itself from Arduino that it has limited functions to interact with external sensors or actuators. Here are some typical projects using Raspberry Pi, a XBMC multimedia server, Wi-Fi enabled home surveillance system, etc.

In the following, the features of Arduino platform and Raspberry Pi are compared and the applicable domains, technical specifications are described. The Arduino Uno R3[8] is picked for comparisons as this is the most popular one among Arduino boards. Table 1 lists the comparison results between Arduino UNO R3 and Raspberry Pi Model B[9]. Due to the fact that there are hundreds of ARM-based C language-supported development boards which will be illustrated later, we only compare Arduino and Raspberry Pi here.

Table 1. Comparisons of Arduino Uno R3 and Raspberry Pi Model B

Name	**Arduino Uno R3**	**Raspberry Pi Model B**
Price	$29.95	$35
Size	2.95”x2.10”	3.37”x2.125”
Processor	ATMega 328 8-bit MCU	ARM 11 32-bit MCU
Clock Speed	16Mhz	700Mhz
RAM	2KB	256MB
Flash	32KB	(SD Card)
EEPROM	1KB	N/A
Input Voltage	7-12V	5V
Min Power	42mA (0.3W)	700mA(3.5W)
Digital GPIO	14	8
Analog Input	6 10-bit Channels	N/A
PWM	6	N/A
TWI/I2C	2	1
SPI	1	1
UART	1	1
Dev IDE	Arduino Tool	IDLE, Scratch, anything with Linux support
Ethernet	Not easy, but doable with shields	10/100 Mb
USB Master	N/A	2 USB 2.0
Video Out	N/A	HDMI, Composite
Audio Output	N/A	HDMI, Analog
Operating System	N/A	Linux distributions
Applicable Domain	Hardware-based with sensors, actuators	Software-based involved internet, multimedia
Multitasking	N/A	Yes

From Table 1, we can see that the price and size of the two devices are comparable and both Raspberry Pi and Arduino are tiny and cheap.

The Raspberry Pi is 40 times faster than an Arduino when it comes to clock speed. Even more seemingly damning for Arduino, Pi has 128,000 times more RAM. The Raspberry

Pi is an independent computer that can run an actual operating system in Linux. It can multitask, support two USB ports, and connect wirelessly to the Internet. In short, it's powerful enough to function as a personal computer (though not powerful enough to compete with your Mac or PC.

It might sound like Raspberry Pi is superior to Arduino, but that's only when it comes to software applications. Arduino's simplicity makes it a much better bet for pure hardware projects. The flexibility of Arduino, availability of around 300 shields allows it to work with just about any kind of sensors or actuators. Raspberry Pi is not flexible to read analog sensors, interface with actuators.

The Arduino IDE is significantly easier to use than Linux. For example, if you wanted to write a program to blink an LED with Raspberry Pi, you'd need to install an operating system and some code libraries—and that's just to start. On Arduino, you can get an LED light to blink in just eight lines of code. Since Arduino isn't designed to run an OS or a lot of software, you can just plug it in and get started.

Raspberry Pi can multitask processes—it can run multiple programs in the background while activated. For example, a Raspberry Pi can be easily programmed to serve as both a print server and a VPN server at the same time.

While the Raspberry Pi shines in software application, the Arduino makes hardware projects very simple. It's simply a matter of figuring out what you want to do.

In summary, the Arduno is more suitable for motor driving, sensor reading, LED driving, etc while you can have an Internet-connected Raspberry Pi , a mini computer that can play videos, music or send emails with ease.

As Arduino platform and Raspberry Pi can find wide applications in students' senior design projects and among hobby electronics projects, there are some limitations of Arduino platforms, such as 1) not suitable to manage concurrent activities in a complex real-time systems, 2) lack of debugging capability which can severely limits the development of complex embedded systems, 3) lack of the support of real-time Operating System., 4) not suitable for time critical embedded system. For Raspberry Pi, it is not suitable for projects interacting sensors and actuators. Therefore, the ARM-based development boards programming with C are playing an important role in the teaching and learning of embedded systems. In the following, we will use a development board MCB1700 from Keil ARM-based company for illustration[10].

The Keil MCB1700 Evaluation Boards enable you to create and test working programs based on the NXP LPC1700 family of ARM Cortex™-M3 processor-based devices.

The MCB1700 board features the following specifications

- 100MHz ARM Cortex-M3 processor-based MCU
- On-Chip Memory: 512KB Flash and 64KB RAM

- Color QVGA TFT LCD
- 10/100 Ethernet Port
- USB 2.0 Full Speed - USB, USB-OTG, and USB Host
- 2 CAN interfaces
- 2 Serial Ports
- SD/MMC Card Interface
- 5-position Joystick and push-button
- Analog Voltage Control for ACD Input
- Amplifier and Speaker
- Up to 70 GPIO
- Debug Interface Connectors

    - 20-pin JTAG (0.1 inch connector)
    - 10-pin Cortex debug (0.05 inch connector)
    - 20-pin Cortex debug + ETM Trace (0.05 inch connector)

The MDK-ARM suite [11] can be used for developing with a MCB1700 board. The MDK-ARM is a complete software development environment for Cortex™-M, Cortex-R4, ARM7™ and ARM9™ processor-based devices. MDK-ARM is specifically designed for microcontroller applications, it is easy to learn and use, yet powerful enough for the most demanding embedded applications. It features

- Complete support for Cortex-M, Cortex-R4, ARM7, and ARM9 devices
- Industry-leading ARM C/C++ Compilation Toolchain
- µVision4 IDE, debugger, and simulation environment
- Keil RTX deterministic, small footprint real-time operating system (with source code)
- TCP/IP Networking Suite offers multiple protocols and various applications
- USB Device and USB Host stacks are provided with standard driver classes
- Complete GUI Library for embedded systems with graphical user interfaces
- Complete Code Coverage information about your program's execution
- Execution Profiler and Performance Analyzer enable program optimization
- Numerous example projects help you quickly become familiar with MDK-ARM's powerful, built-in features
- CMSIS Cortex Microcontoller Software Interface Standard compliant

Other real-time operating system such as uC/OS-II, uC/OS-III, Free RTOS can also be easily ported to MCB1700 board in Keil MDK environments. Complex embedded system with critical time requirements can be developed with MCB1700 and Keil MDK suite, however, the learning curve is a little deep compared to Arduino and Raspberry Pi and the commercial price for the license ($5000 per license) is much higher compared to Raspberry Pi and Arduino platforms.

4. SUMMARY

In this paper, the popular embedded system platforms – Arduino, Raspberry Pi and ARM-based development boards are studied and compared. Arduino platform can be used in the hardware-based projects with sensors, actuators, etc. Raspberry Pi can find popular applications in software-based projects involving graphics, internet. The ARM-based development boards with commercial or open-source toolchains and with the support of powerful debugging capabilities, real-time operating systems can be used in complex time critical embedded system and play an important role in the teaching and learning of embedded systems in the classroom.

Reference

1. Raspberry Pi website, http://www.raspberry pi.org
2. Arduino official website, http://www.arduino.cc
3. John Wharton: An Introduction to the Intel MCS-51 Single-Chip Microcomputer Family, Application Note AP-69, May 1980, Intel Corporation.
4. Tom Krazit, "ARMed for the living room", CNet.com, April 3 2006.
5. Dan Grabham, "From a small Acorn to 37 billion chips: ARM's ascent to tech superpower", TechRadar, July 19 2013.
6. J. Fitzpatrick, "An interview with Steve Furber", Communications of the ACM, Vol 54, No 5, 2011.
7. Arduino Shields List, at http://shieldlist.org/
8. Arduino Uno R3, http://arduino.cc/en/Main/arduinoBoardUno
9. Raspberry Pi Model B, http://www.raspberrypi.org/products/model-b/
10. Keil MCB1700 Evaluation Board overview, http://www.keil.com/mcb1700/
11. Keil MDK-ARM Suite, http://www.keil.com/arm/mdk.asp